

日本国特許庁
JAPAN PATENT OFFICE

3/ 01-201

JC879 U.S. PTO
10/046982



#2
2-26-02
all

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application:

2001年 4月12日

出願番号
Application Number:

特願2001-113931

出願人
Applicant(s):

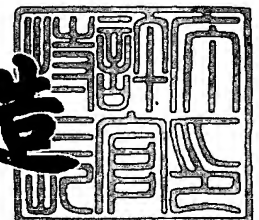
株式会社日立製作所

U.S. Appln. Filed 1-17-02
Inventor: T. Nakamura
Mattingly Stanger & Malor
Docket H-1021

2001年10月19日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造



出証番号 出証特2001-3092551

【書類名】 特許願

【整理番号】 H100020I

【提出日】 平成13年 4月12日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 9/38

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社
日立製作所 中央研究所内

【氏名】 中村 友洋

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社日立製作所

【代理人】

【識別番号】 100099298

【弁理士】

【氏名又は名称】 伊藤 修

【連絡先】 0 3 - 3 2 5 1 - 3 8 2 4

【選任した代理人】

【識別番号】 100099302

【弁理士】

【氏名又は名称】 笹岡 茂

【手数料の表示】

【予納台帳番号】 018647

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 プロセッサおよびプロセッサシステム

【特許請求の範囲】

【請求項1】 命令の処理に先立ってもしくはは並行してその命令の処理結果の値を予測し、予測された結果を使用して仮に後続の処理を行うことを許可する印をつけられるフィールドを備える命令を実行する手段を有することを特徴とするプロセッサ。

【請求項2】 命令の処理に先立ってもしくはは並行してその命令の処理結果の値を予測し、予測された結果を使用して仮に後続の処理を行うことを許可する印をつけられるフィールドと、命令の処理結果の値を予測する方法を指定するフィールドを備える命令を実行する手段を有することを特徴とするプロセッサ。

【請求項3】 命令の処理に先立ってもしくはは並行してその命令の処理結果の値を予測し、予測された結果を使用して仮に後続の処理を行うことを許可する印をつけられるフィールドと、命令の処理結果の値を予測し、その結果をその真偽に関わらず常に使用して処理を行うことを許可する印をつけられるフィールドを備える命令を実行する手段を有することを特徴とするプロセッサ。

【請求項4】 命令の処理に先立ってもしくはは並行してその命令の処理結果の値を予測し、予測された結果を使用して仮に後続の処理を行うことを許可する印をつけられるフィールドと、命令の処理結果の値を予測する方法を指定するフィールドと、命令の処理結果の値を予測し、その結果をその真偽に関わらず常に使用して処理を行うことを許可する印をつけられるフィールドを備える命令を実行する手段を有することを特徴とするプロセッサ。

【請求項5】 命令を格納するキャッシュメモリを備えるプロセッサにおいて、

該キャッシュメモリに格納された命令毎に、その命令の処理に先立ってもしくはは並行してその命令の処理結果の値を予測し、予測された結果を使用して仮に後続の処理を行うことを許可する印をつけられるフィールドを備えることを特徴とするプロセッサ。

【請求項6】 請求項5に記載のプロセッサにおいて、

前記キャッシュメモリに格納された命令毎に、命令の処理結果の値を予測する方法を指定するフィールドを備えることを特徴とするプロセッサ。

【請求項 7】 請求項 5 もしくは請求項 6 に記載のプロセッサにおいて、命令の処理結果の値を予測した値を使用して実行した結果の真偽に応じて前記キャッシュメモリに格納された命令毎に用意されたフィールドの値を更新する手段を有することを特徴とするプロセッサ。

【請求項 8】 請求項 5 もしくは請求項 6 もしくは請求項 7 に記載のプロセッサにおいて、

前記キャッシュメモリに格納された命令毎に、命令の処理結果の値を予測し、その結果をその真偽に関わらず常に使用して処理を行うことを許可する印をつけられるフィールドを備えることを特徴とするプロセッサ。

【請求項 9】 命令の処理に先立ってもしくは並行してその命令の処理結果の値を予測し、予測された結果（予測値）を使用して仮に後続の処理を行うことを許可する印をつけられる第 1 のフィールドと、命令の処理結果の値を予測する方法を指定する第 2 のフィールドを備える命令が格納されるキャッシュメモリと

予測値を出力する複数の値予測手段と、

予測値を格納する格納手段と、

命令の処理結果と該命令の処理結果の予測値が一致するか否かを判定する判定手段と、

前記キャッシュメモリから読み出された命令の実行と、前記格納手段に格納された予測値を用いた後続命令の実行を行う演算手段を有し、

前記命令の第 1 のフィールドに前記許可する印がつけられているとき、該命令の第 2 のフィールドにより指定された値予測手段の予測値を前記格納手段と前記判定手段に出力し、

該命令に対する該判定手段の判定結果が不一致のとき、前記キャッシュメモリ中の該命令の第 1 のフィールド及び第 2 のフィールドのデータを更新することを特徴とするプロセッサ。

【請求項 10】 コンピュータに、命令の処理に先立ってもしくは並行して

その命令の処理結果の値を予測し予測された結果を使用して仮に後続の処理を行うことを許可する印をつけられるフィールドを備える命令を有するプログラムに対し、該フィールドの印を許可もしくは不許可に設定する機能を実現させるためのプログラム。

【請求項 11】 請求項 10 記載のプログラムにおいて、

前記命令は、さらに命令の処理結果の値を予測する方法を指定するフィールドを備え、該フィールドに値を予測する方法を設定する機能を実現させるためのプログラム。

【請求項 12】 請求項 10 もしくは請求項 11 記載のプログラムにおいて

前記命令は、さらに命令の処理結果の値を予測しその結果をその真偽に関わらず常に使用して処理を行うことを許可する印をつけられるフィールドを備え、該フィールドの印を許可もしくは不許可に設定する機能を実現させるためのプログラム。

【請求項 13】 命令の処理に先立ってもしくは並行してその命令の処理結果の値を予測し予測された結果を使用して仮に後続の処理を行うことを許可する印をつけられるフィールドを備える命令を有するプログラムの該各命令のフィールドの印を許可もしくは不許可に設定する設定手段と、該設定の行われた各命令を実行する実行手段を有することを特徴とするプロセッサシステム。

【請求項 14】 請求項 13 記載のプロセッサシステムにおいて、

前記プログラムの命令は、さらに命令の処理結果の値を予測する方法を指定するフィールドを備え、前記設定手段は、さらに該フィールドに値を予測する方法を設定し、前記実行手段は全ての設定の行われた各命令を実行することを特徴とするプロセッサシステム。

【請求項 15】 請求項 13 もしくは請求項 14 記載のプログラムにおいて

前記プログラムの命令は、さらに命令の処理結果の値を予測しその結果をその真偽に関わらず常に使用して処理を行うことを許可する印をつけられるフィールドを備え、前記設定手段は、さらに該フィールドの印を許可もしくは不許可に設

定し、前記実行手段は全ての設定の行われた各命令を実行することを特徴とするプロセッサシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、高性能コンピュータ・システムに係り、特に、命令の処理結果の値を予測し、予測結果を用いて後続命令の処理を行うコンピュータシステムに関する。

【0002】

【従来の技術】

高性能コンピュータ・システムにおいて、プログラムの実行時間を短縮するために、命令の処理順序を規定する制御依存関係とデータ依存関係をプログラムの結果が正しくなる範囲でできる限り実行時間が短くなるように並び替えることが行われている。

制御依存関係については、分岐命令の結果を予測することで、分岐方向が決定する前に、予測した分岐先の命令の処理を進めることが行われている。

これにより予測が成功すれば、データ依存関係のない命令は分岐命令の処理の終了にかかわらず処理可能となり、処理の高速化がはかられる。

予測が失敗した場合は、予測により実行された処理を取り消し、正しい分岐先の命令を再実行することでプログラムの実行結果を正しく保てる機構をもつ。

一方データ依存関係については、演算結果やデータ転送命令によりロードされるデータの値を予測することで処理性能の向上をはかる方法がある。

値予測の方法としては、前回と同一の値が繰り返されると予測する最終値予測、一定の差分でデータ値が変化していくと予測するストライド値予測、過去の値の振る舞いのパターンに対する相関を利用する2レベル値予測などの方式が知られている。

【0003】

【発明が解決しようとする課題】

値予測によるデータ依存関係の解消は、値が正しく予想されれば性能向上をは

かれるが、間違った値を予測すると、値を予測しなかった場合に比べて性能が低下する可能性がある。

また、値の予測には通常、過去の履歴に基づく方式が利用されるため、命令によって値予測の成功率に大きな差があるのが通常であり、値予測の成功率の高いものにのみ選択的に値予測を行う必要がある。

本発明の目的は、値予測を行った方が処理性能が向上すると思われる命令を指定できるようにすることにある。

本発明の他の目的は、値予測を行う際の予測精度を向上することにある。

【0004】

【課題を解決するための手段】

上記目的を達成するため、本発明では、命令の一部に値予測を行うべきかどうかを指示するフィールドを設け、もしくは、命令キャッシュメモリに値予測を行うべきかどうかを指示するフィールドおよび値予測方式を指定するフィールドを併設し、値予測を行った方が処理性能が向上すると思われる命令を指定し、この命令をプロセッサで実行することにより、精度の高い予測を行うようにしている。

また、さらに、命令に値予測の結果を常に使用することを指示するフィールドを設け、この命令をプロセッサで実行することにより、処理の高速化をはかるようにしている。

また、上記のフィールドを備える命令からなるプログラムの内容にしたがって、該命令のフィールドの内容を設定するようにしている。

【0005】

【発明の実施の形態】

本発明による命令フォーマットの1実施例は図1に示す。

命令はオペコード(1)およびディスティネーションレジスタRD(2)、第1ソースレジスタRS1(3)、第2ソースレジスタRS2(4)に加えて、値予測フィールド(5)をもつ。

値予測フィールド(5)は、処理結果を予測する必要のある命令にのみ値予測フィールド(5)があればよく、すべての命令セットに必要であるというもので

はない。

値予測フィールド（５）は１ビットからなり、このビットが１ならば値予測を行い、０ならば値予測を無効化する。

図２は図１の命令フォーマットに追加して、さらに値予測方式フィールド（６）を備えたものである。

値予測方式フィールドはＮ個（Ｎは１以上の整数）の値予測方式から１つを選択するものである。

値予測方式フィールドも、処理結果を予測する必要がある命令で、さらに値予測器が複数ある場合で、その中の１つがその命令に対してより高い確率で値予測が可能である場合に、命令に付加されものであり、すべての命令セットに必要であるというものでない。

【 0 0 0 6 】

本発明による、値予測フィールド（５）および値予測方式フィールド（６）をそなえた命令をもつコンピュータシステムの構成図を図３に示し、コンピュータシステムの動作をこの図３により説明する。

コンピュータシステムは、それぞれ値予測の方式が異なるＮ個の値予測器（１２）、セレクタ１３、値予測マスクゲート（１４）、演算器（１５）、レジスタ（１６）を備える。

値予測フィールド（５）および値予測方式フィールド（６）を除いた命令の実行は演算器（１５）で行われる。

それぞれ値予測の方式が異なるＮ個の値予測器（１２）によるそれぞれの値予測結果のうち、値予測方式フィールド（６）により指定される値予測器の予測結果がセレクタ（１３）で選択される。

選択された予測値は値予測フィールド（５）と値予測マスクゲート（１４）で論理積がとられ、それが有効もしくは無効にされる。

有効となった予測値はレジスタ（１６）に書き込まれ、後続の命令により演算器（１５）で利用される。

ただし、演算器（１５）による処理結果が予測値と異なった場合は、予測値が書き込まれたレジスタ（１６）は無効化され、その予測値を使って処理された後

続命令もキャンセルされる。

また、値予測フィールド（５）および値予測方式フィールド（６）に格納されたデータをオペコード（１）の中に符号化して組み込むこともでき、その場合命令フォーマット中には値予測フィールド（５）や値予測方式フィールド（６）は存在しなくなるが、オペコード（１）に含まれることになる。

【0007】

本発明の他の実施例は図４に示した命令キャッシュメモリ（２０）である。

この命令キャッシュメモリ（２０）は命令フィールド（７）に加えて、値予測フィールド（８）をもつ。

値予測フィールドは図１の値予測フィールド（５）と同様の意味を持つ。

図５は図４に加えて、命令キャッシュメモリ（２０）の各命令に対して、値予測方式フィールド（９）を追加したものである。

この値予測方式フィールド（９）も図２の値予測方式フィールド（６）と同様の意味を持つ。

【0008】

図５の命令キャッシュメモリ（２０）をもつコンピュータシステムの構成を図６に示し、このコンピュータシステムにおける命令処理手順をこの図６により説明する。

コンピュータシステムは、それぞれ値予測の方式が異なるＮ個の値予測器（１２）、セレクタ１３、値予測マスクゲート（１４）、演算器（１５）、レジスタ（１６）、値予測結果判定器（１７）、命令キャッシュメモリ（２０）を備える。

値予測フィールド（８）および値予測方式フィールド（９）を除いた命令の実行は演算器（１５）で行われる。

それぞれ値予測の方式が異なるＮ個の値予測器（１２）のそれぞれの値予測結果のうち、命令キャッシュメモリ（２０）から読み出された命令の値予測方式フィールド（９）により指定される値予測器の予測結果がセレクタ（１３）で選択される。

選択された予測値は値予測フィールド（８）と値予測マスクゲート（１４）で

論理積がとられ、選択された予測値は有効もしくは無効にされる。

有効となった予測値はレジスタ（１６）に書き込まれ、後続の命令により演算器（１５）で利用される。

ただし、演算器（１５）による処理結果が予測値と異なった場合は、予測値が書き込まれたレジスタ（１６）は無効化され、その予測値を使って処理された後続命令もキャンセルされる。

演算器（１５）と予測値の比較は値予測結果判定器（１７）により判定され、その結果は命令キャッシュメモリ（２０）に通知されて、該予測値に対応する命令キャッシュメモリ（２０）内の命令の値予測フィールド（８）および値予測方式フィールド（９）のデータが更新される。これにより、命令キャッシュメモリ（２０）内の命令が再度実行されるとき、更新された命令が実行されることになる。

【 0 0 0 9 】

本発明が対象とする値予測器の実施例としては、図 7、図 8、図 9 が挙げられる。

図 7 の値予測器は最終値予測方式によるものであり、命令を特定するための値であるプログラムカウンタなどのインデックス値により値予測表のタグ（２１）を引き、該当するインデックスの前の値（Last Value：最終値）（２２）を予測値とするものである。

この値予測器では、同じ Tag に対する予測値が常に同じ値をとる値、即ち、予測値が一定値をとる値、については正しく予測されるが、それ以外の値、即ち、予測値が常に同じ値とは言えない場合では正しく予測されないことがある。ただし、この値予測器では、ハードウェア量が少ないため実装が簡単である。

なお、図において、0 x 2 0 2 0 等における 'x' は 1 6 進数を表す。

【 0 0 1 0 】

図 8 は図 7 に示した値予測器における最終値（Last Value）（２２）にインターバル値（Interval）（２３）を加えたものを予測値とする値予測器である。即ち、Last Value を L とし、Interval を I とした場合、 $L = L \pm I$ により、L の値が変化してゆく。。

この値予測器では、図7の値予測器で正しく予測が可能な定数値の場合に加えて、一定のインターバルで値が増減するものについても正しく値予測できる。

ただし、インターバル値が一定で無い場合には正しく予測されないことがある。

この値予測器では、ハードウェア量は図7の値予測器に比べて増える。

【0011】

図9は2レベルの予測方式による値予測器であり、インデックス値による値予測表のタグ(21)を引きオフセット値(Offset Value)(24)と履歴情報(History)(25)を取り出す。

次に履歴情報(25)で履歴タグ(History Tag)(26)を引き、リンク値(Link Value)(27)を得る。

予測値はオフセット値(24)とリンク値(27)を加えたものとなる。

履歴情報(25)は、予測結果の正解、不正解により更新されるようにしてもよい。例えば、履歴情報(25)が‘0000’の場合、正解なら‘0000’とし、不正解なら‘0001’とする。

また、リンク値(27)も予測結果の正解、不正解により更新されるようにしてもよい。オフセット値(24)=1で、リンク値(27)=0なら予測値は‘1’であるが、実際の演算結果が‘2’である場合には、リンク値(27)を‘1’更新する。

この値予測器では、図7や図8の値予測器で正しく予測が可能な場合には、ほとんどの場合正しく予測でき、さらに通常は定数値であるが、ある間隔で異なった値に変化するケースなどで正しく予測が可能な場合がある。

図7や図8に示した値予測器に比べてハードウェア量は大きいが正しく予測される確率は一般的に高い特徴を持つ。

【0012】

図7、図8、図9で示した値予測器の実装例は、図3や図6における値予測器1～Nの実装例である。

図3や図6では、これらの値予測器の中から、最終的な値予測を行う値予測器を値予測ごとに選択できる。

【0013】

図10、図11、図12は値予測が効果的に働く例（プログラムも表示）を示す。

図10に示した例では、アドレス0x2020においてレジスタr1の値が0に設定され、アドレス0x2024から0x2030までのループ処理において、r1の値は常に0となる。

またアドレス0x2024においてレジスタr2の値は常にレジスタr1に4を加えた値となるので、ループ処理中は常に4である。

同様にしてアドレス0x2028においてレジスタr3の値は常に8、アドレス0x202Cにおいてレジスタr4の値は常に12である。

よってアドレス0x2020、0x2024、0x2028、0x202Cのすべての命令の演算結果は常に定数となり、これは図7に示した値予測器によって正しく値予測が行われる。

【0014】

図11に示した例では、アドレス0x2024においてレジスタr1の値がループ処理ごとに1ずつ増える。

アドレス0x2028においてレジスタr2の値はレジスタr1の値に4を加えた値となるので、ループ処理ごとに1ずつ増える。

同様にしてアドレス0x202Cにおいてレジスタr3の値はループ処理ごとに1ずつ増え、アドレス0x2030においてレジスタr4の値もループ処理ごとに1ずつ増える。

図11に示した例は、図7に示した値予測器によって正しく予測することはできないが、図8に示した値予測器によって正しく予測することができる。

【0015】

図12に示した例では、アドレス0x2028においてレジスタr1の値がループ処理ごとに-1と+1を交互にとる。

アドレス0x202Cにおいてレジスタr2の値はレジスタr1の値に4を加えた値となるので、ループ処理ごとに3と5を交互にとる。

同様にしてアドレス0x2030においてレジスタr3の値はループ処理ごと

に7と9を交互にとり、アドレス0x2034においてレジスタr4の値はループ処理ごとに11と13を交互にとる。

図12に示した例は、図7や図8に示した値予測器によって正しく予測することはできないが、図9に示した値予測器によって正しく予測することができる。

【0016】

図9の値予測器によって図12のプログラムの値予測を行った場合の、定常状態における値予測器の状態は図13に示したようになる。

図13において、Tagが0x2028のとき、オフセット値(24)=-1であり、履歴情報(25)は'0101'で、リンク値(27)=2となり、予測値は'1'となる。

予測値が正になる場合には、履歴情報(25)の'0101'の末尾に'0'を追加して左シフトし、履歴情報(25)を'1010'にする。

この次にTagが0x2028になったとき、履歴情報(25)は'1010'であるため、リンク値(27)=0となり、予測値は'-1'となり、予測値は負になる。

予測値が負になる場合には、履歴情報(25)の'1010'の末尾に'1'を追加して左シフトし、履歴情報(25)を'0101'にする。

そこで、この次にTagが0x2028になったとき、履歴情報(25)は'0101'であるため、リンク値(27)=2となり、予測値は'1'となり、予測値は正になる。

これにより、Tagが0x2028になる度に、予測値は'1'と'-1'の間で切り替わる。

【0017】

図10、図11、図12に示した例では、コンパイラプログラムがプログラムにおけるデータ依存関係を解析することにより、レジスタr1、r2、r3、r4の値が決定できるため、値予測器を使用しなくても処理速度を向上させられる。

しかし、図14、図15、図16に示すように図10、図11、図12におけるそれぞれのアドレス0x2020、0x2020、0x2024の命令がロー

ド命令に置き換わった場合には、レジスタ r_1 の値がコンパイラプログラムによる静的な解析では決定できないため、コンパイラプログラムによって処理速度を向上させることは困難である。

そこで、本発明による値予測をすべき命令としてのマーク（すなわち、図1または図2の命令フォーマットの値予測フィールドの値としての‘1’）を図14では $0 \times 2024 \sim 0 \times 202C$ 、図15では $0 \times 2024 \sim 0 \times 2030$ 、図16では $0 \times 2028 \sim 0 \times 2034$ の命令に対して付けることにより、値予測器によって正しく値予測が行われる可能性が高いことを示し、実際に値予測が正しく行われれば処理速度の向上が達成される。

【0018】

図1もしくは図2に示した命令をもつプロセッサの実行コードを作成するコード作成用プログラムでは、図14、図15、図16のプログラムを次のように解析して、値予測フィールド（5）、値予測方式（6）に値を設定する。

図14では、レジスタ r_1 の値がプログラム中のループの外側にあるアドレス 0×2020 の命令で設定されており、ループの中では値が変更されないことがコード作成用プログラムによって解析することで分かる。

さらに、 0×2024 、 0×2028 、 $0 \times 202C$ の各 `add` 命令では、それぞれレジスタ r_1 の値に定数 0×4 、 0×8 、 $0 \times C$ を加算しているため、これらの加算結果が格納されるレジスタ r_2 、 r_3 、 r_4 はループ内で不変の値となることが分かる。

よって、前回の値を次回の値予測値とする図7の最終値予測方式の値予測器によって正しく値を予測することができるので、値予測フィールド（5）を値予測を有効とする意味を持つ1に設定し、値予測方式フィールド（6）には図7の最終値予測方式を指定する値をコード作成用プログラムによって設定する。

【0019】

図15では、プログラム中のループの内側にあるアドレス 0×2024 の命令でレジスタ r_1 の値が毎回1ずつ加算されていることがコード作成用プログラムによって解析することで分かる。

よって、毎回一定値を加算して値予測値とする図8のインターバル値予測方式

の値予測器によって正しく値を予測することができるので、アドレス0x2024の命令の値予測フィールド(5)を値予測を有効とする意味を持つ1に設定し、値予測方式フィールド(6)には図8のインターバル予測方式を指定する値をコード作成用プログラムによって設定する。

さらに、アドレス0x2028、0x202C、0x2030の各命令では、毎回一定値が加算されていくレジスタr1の値に定数値0x4、0x8、0xCをそれぞれ加算したデータをレジスタr2、r3、r4に格納していることがコード作成用プログラムによって解析することで分かる。

よってr2、r3、r4のデータも毎回一定値を加算して値予想値とする図8のインターバル値予測方式の値予測器によって正しく値を予測することができるので、アドレス0x2028、0x202C、0x2030の各命令の値予測フィールド(5)も値予測を有効とする意味を持つ1に設定し、値予測方式フィールド(6)には図8のインターバル値予測方式を指定する値をコード作成用プログラムによって設定する。

【0020】

図16では、プログラム中のループの内側にあるアドレス0x2028の命令でレジスタr1の値が交互に正負の値を繰り返し取ることがコード作成用プログラムによって解析することで分かる。

よって、前回までの値の履歴によって次の予測する値が決定される図9の2レベルの値予測方式の値予測器によって正しく値を予測することができるので、アドレス0x2028の命令の値予測フィールド(5)を値予測を有効とする意味を持つ1に設定し、値予測方式フィールド(6)には図9の2レベル値予測方式を指定する値をコード作成用プログラムによって設定する。

さらに、アドレス0x202C、0x2030、0x2034の各命令では、交互に値の正負が変化するレジスタr1の値に定数値0x4、0x8、0xCをそれぞれ加算したデータをレジスタr2、r3、r4に格納していることがコード作成用プログラムによって解析することで分かる。

よってr2、r3、r4のデータもr1のデータと同様にして図9の2レベル値予測方式の値予測器によって正しく値を予測することができるので、アドレス

0x202C、0x2030、0x2034の各命令の値予測フィールド（5）も値予測を有効とする意味を持つ1に設定し、値予測方式フィールド（6）には図9の2レベル値予測方式を指定する値をコード作成用プログラムによって設定する。

【0021】

メディア処理など、処理の内容によっては、演算結果の真偽において精度が低いことが許される場合がある。

そのような場合には、演算結果を値予測によって決定して後続の処理を進めることで、処理速度の向上を実現することが可能である。

【0022】

図17は値予測の結果を常に使用するためのフィールドを備えた命令の実施例である。

図17で強制値予測（28）フィールドによって、この命令の処理結果は、値予測ができた場合には常に値予測結果を信用して、これを使用して後続の処理を継続する。

図18は、これと同様の機能を命令キャッシュに強制値予測（29）フィールドを備えることで実現した実施例である。

図19は、本発明による値予測を行うコンピュータシステムの一実施例である。

プログラムコード（30）は本発明の値予測関係フィールドをもった命令列を出力するコンパイラプログラム（31）によってコンパイルされ、この実行形式コードが命令キャッシュ（20）に登録される。

命令キャッシュ（20）からコードが実行されると、値予測方式フィールド（9）により値予測器が選択（12）され、さらに値予測フィールド（8）の値に応じてその結果の使用が決定（14）される。

一方この命令の演算は演算器（15）において並行して行われ、その結果と値予測の結果が比較（17）され、この結果に応じて命令中のフィールド値が更新（18）される。

【0023】

【発明の効果】

本発明によれば、値予測により性能向上がはかれる命令にのみ値予測を適用するための命令もしくは命令キャッシュメモリを用意することで、不必要な値予測による性能低下を押さえることができる。

また、より精度の高い値予測を行うのに必要な、値予測方式の選択を命令毎に可能とすることで、値予測を利用したコンピュータシステムの性能向上率を高めることができる。

【図面の簡単な説明】

【図 1】

本発明による値予測フィールドをもつ命令フォーマットを示す図である。

【図 2】

本発明による値予測フィールドおよび値予測方式フィールドをもつ命令フォーマットを示す図である。

【図 3】

本発明による図 2 に示す命令フォーマットをもつコンピュータシステムの構成を示す図である。

【図 4】

本発明による値予測フィールドをもつ命令キャッシュを示す図である。

【図 5】

本発明による値予測フィールドおよび値予測方式フィールドをもつ命令キャッシュを示す図である。

【図 6】

本発明による図 5 に示す命令キャッシュをもつコンピュータシステムの構成を示す図である。

【図 7】

本発明による値予測器の実施例を示す図である。

【図 8】

本発明による値予測器の他の実施例を示す図である。

【図 9】

本発明による値予測器のさらに他の実施例を示す図である。

【図 1 0】

値予測効果がある 1 番目のプログラム例を示す図である。

【図 1 1】

値予測効果がある 2 番目のプログラム例を示す図である。

【図 1 2】

値予測効果がある 3 番目のプログラム例を示す図である。

【図 1 3】

本発明による値予測器の別の実施例を示す図である。

【図 1 4】

値予測効果がある 4 番目のプログラム例を示す図である。

【図 1 5】

値予測効果がある 5 番目のプログラム例を示す図である。

【図 1 6】

値予測効果がある 6 番目のプログラム例を示す図である。

【図 1 7】

本発明による値予測フィールドおよび値予測方式フィールドおよび強制値予測フィールドをもつ命令フォーマットを示す図である。

【図 1 8】

本発明による値予測フィールドおよび値予測方式フィールドおよび強制値予測フィールドをもつ命令キャッシュを示す図である。

【図 1 9】

本発明によるコンピュータシステムの構成を示す図である。

【符号の説明】

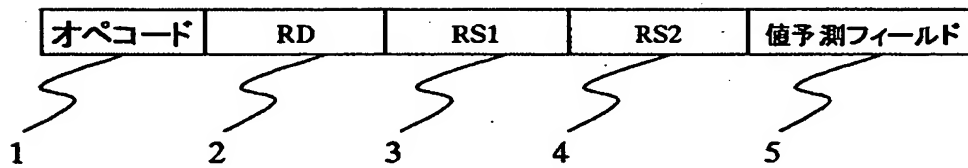
- 5、8 値予測フィールド
- 6、9 値予測方式
- 1 0 値予測イネーブル信号線
- 1 1 値予測方式セレクト信号線
- 1 2 値予測器

- 13 値予測値セレクタ
- 14 値予測マスクゲート
- 15 演算器
- 16 レジスタ
- 17 値予測結果判定器
- 18 値予測結果判定信号線
- 30 プログラムコードを格納するディスク装置
- 31 コンパイラプログラム
- 32~35 命令コード転送線

【書類名】 図面

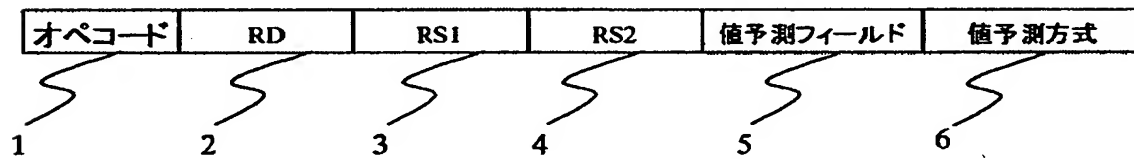
【図1】

【図1】

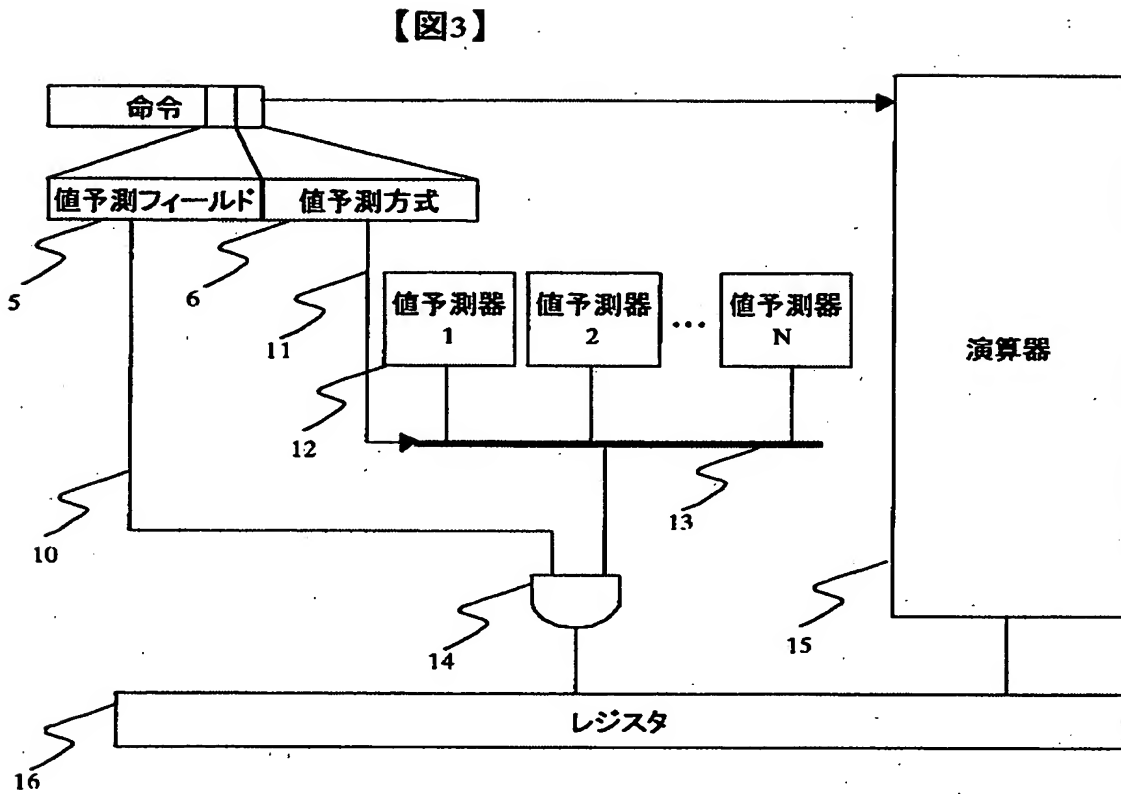


【図2】

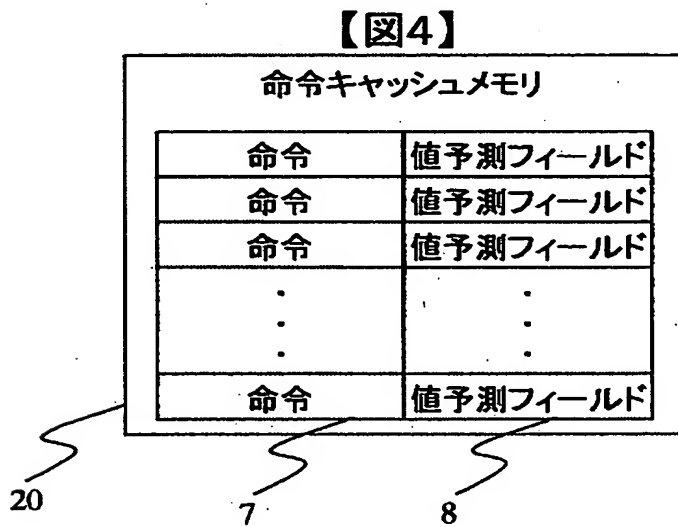
【図2】



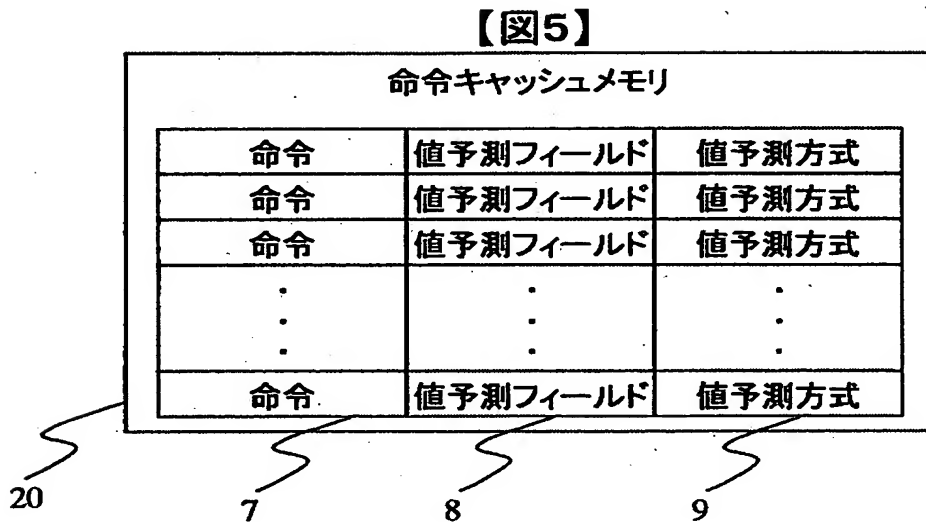
【図3】



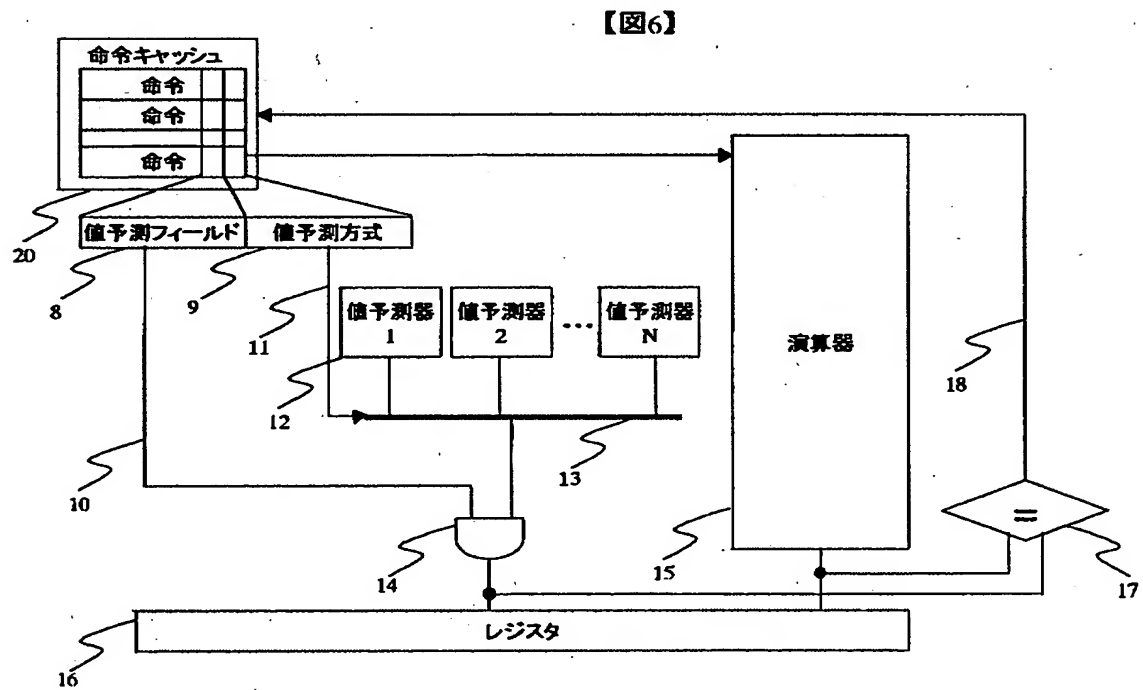
【図4】



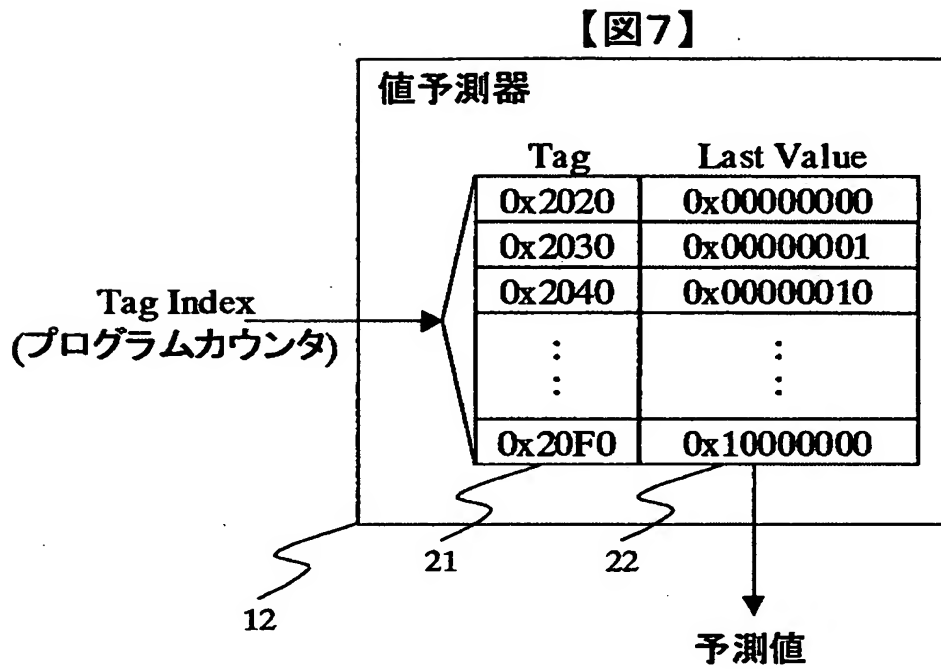
【図5】



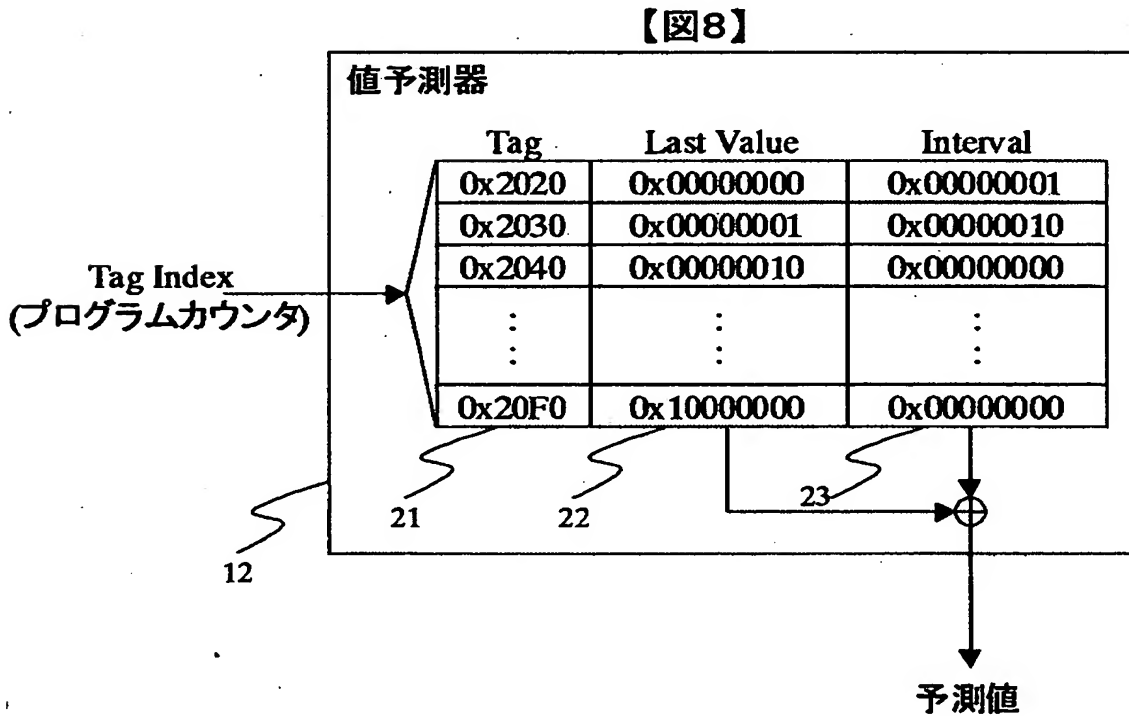
【図6】



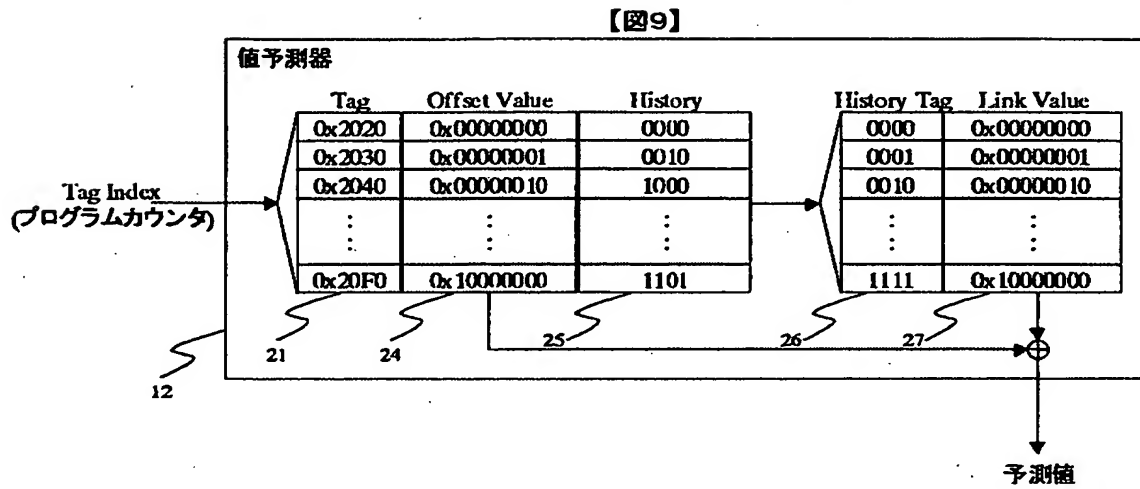
【図 7】



【図 8】



【図 9】



【図 1 0】

【図10】

0x2020	xor r1, r1, r1	: r1=0	r1は常に0
loop:			
0x2024	add r2, r1, 0x4	: r2=r1+4=4	r2は常に4
0x2028	add r3, r1, 0x8	: r3=r1+8=8	r3は常に8
0x202C	add r4, r1, 0xC	: r4=r1+12=12	r4は常に12
0x2030	bc loop	: loop	

【図 1 1】

【図11】

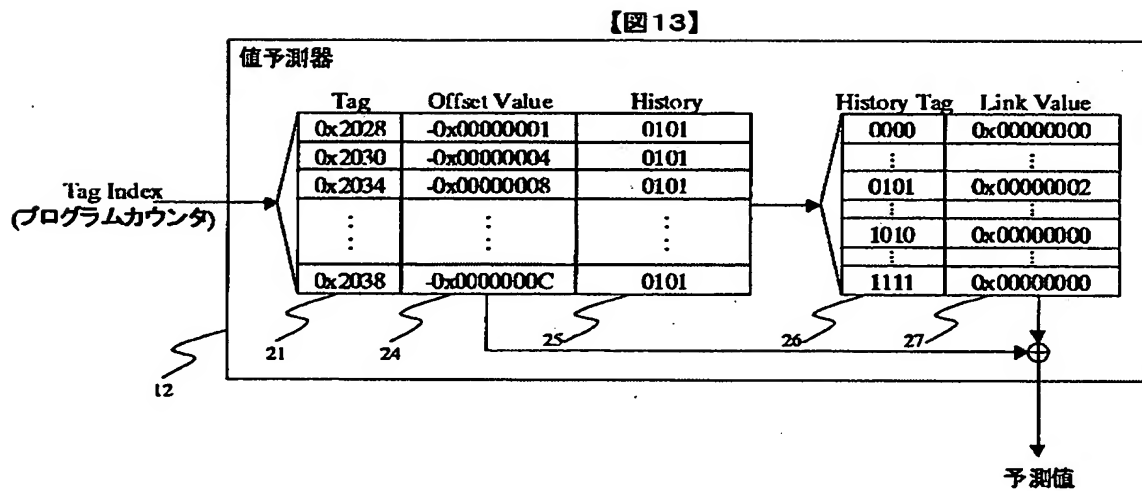
0x2020	xor r1, r1, r1	: r1=0	r1の初期値は0
	loop:		
0x2024	add r1, r1, 0x1	: r1=r1+1	r1は毎回1増える
0x2028	add r2, r1, 0x4	: r2=r1+4	r2は毎回1増える
0x202C	add r3, r1, 0x8	: r3=r1+8	r3は毎回1増える
0x2030	add r4, r1, 0xC	: r4=r1+12	r4は毎回1増える
0x2034	bc loop	: loop	

【図 1 2】

【図12】

0x2020	xor r1, r1, r1	: r1=0	
0x2024	add r1, r1, 0x1	: r1=r1+1=1	r1の初期値は1
	loop:		
0x2028	mul r1, r1, -0x1	: r1=r1 × (-1)	r1は交互に-1,1
0x202C	add r2, r1, 0x4	: r2=r1+4	r2は交互に3,5
0x2030	add r3, r1, 0x8	: r3=r1+8	r3は交互に7,9
0x2034	add r4, r1, 0xC	: r4=r1+12	r4は交互に11,13
0x2038	bc loop	: loop	

【図 13】



【図 14】

【図 14】

<p>0x2020 ld r1, adr</p> <p>loop:</p> <p>0x2024 add r2, r1, 0x4</p> <p>0x2028 add r3, r1, 0x8</p> <p>0x202C add r4, r1, 0xC</p> <p>0x2030 bc loop</p>	<p>: r1 ← Mem[adr]</p> <p></p> <p>: r2 = r1 + 4 = 4</p> <p>: r3 = r1 + 8 = 8</p> <p>: r4 = r1 + 12 = 12</p> <p>: loop</p>	<p>r1にメモリから</p> <p>値をロード</p> <p>r2は常に4</p> <p>r3は常に8</p> <p>r4は常に12</p>
---	---	---

【図15】

【図15】

0x2020	ld r1, adr	: r1←Mem[adr]	r1にメモリから 値をロード
	loop:		
0x2024	add r1, r1, 0x1	: r1=r1+1	r1は毎回1増える
0x2028	add r2, r1, 0x4	: r2=r1+4	r2は毎回1増える
0x202C	add r3, r1, 0x8	: r3=r1+8	r3は毎回1増える
0x2030	add r4, r1, 0xC	: r4=r1+12	r4は毎回1増える
0x2034	bc loop	: loop	

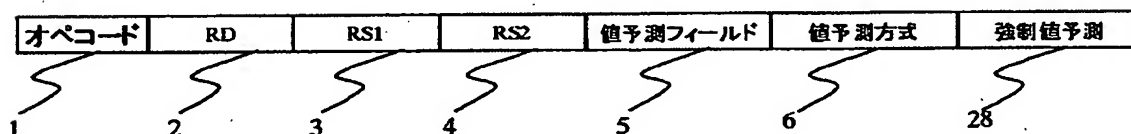
【図16】

【図16】

0x2020	xor r1, r1, r1	: r1=0	
0x2024	ld r1, adr	: r1←Mem[adr]	r1にメモリから 値をロード
	loop:		
0x2028	mul r1, r1, -0x1	: r1=r1×(-1)	r1は交互に-1,1
0x202C	add r2, r1, 0x4	: r2=r1+4	r2は交互に3,5
0x2030	add r3, r1, 0x8	: r3=r1+8	r3は交互に7,9
0x2034	add r4, r1, 0xC	: r4=r1+12	r4は交互に11,13
0x2038	bc loop	: loop	

【図17】

【図17】



【図18】

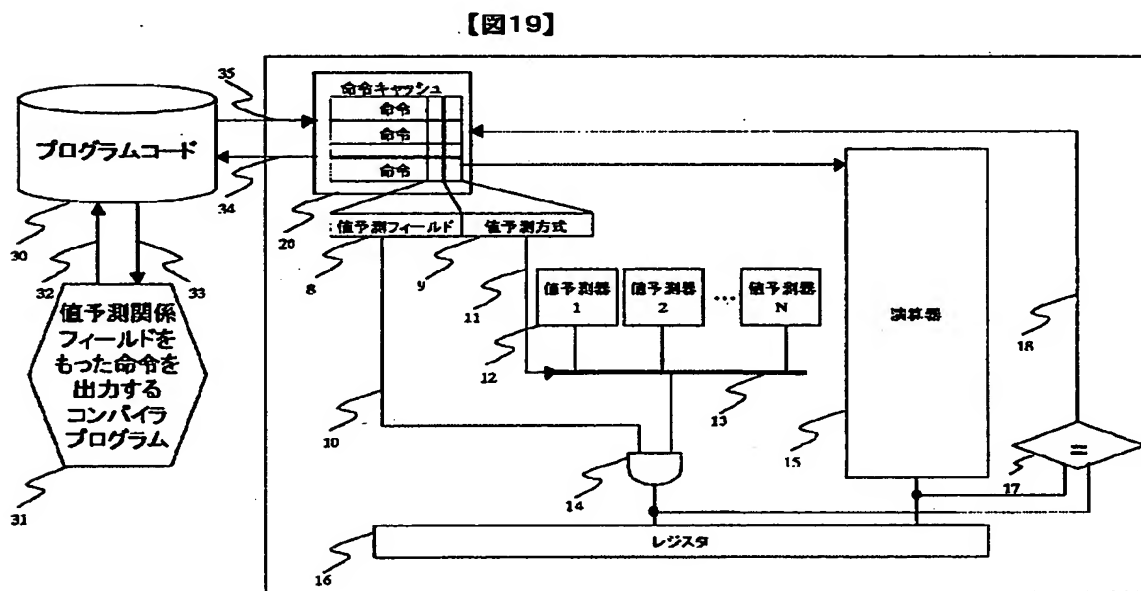
【図18】

命令キャッシュメモリ

命令	値予測フィールド	値予測方式	強制値予測
命令	値予測フィールド	値予測方式	強制値予測
命令	値予測フィールド	値予測方式	強制値予測
⋮	⋮	⋮	⋮
命令	値予測フィールド	値予測方式	強制値予測

20 7 8 9 29

【図19】



【書類名】 要約書

【要約】

【課題】 値予測を行った方が処理性能が向上すると思われる命令を指定して実行できるようにし、また、値予測を行う際の予測精度を向上する。

【解決手段】 命令キャッシュ20には、値予測フィールド8と値予測方式フィールド9を備える命令が格納されている。命令が読み出され、演算器15で実行されると同時に、命令の値予測方式フィールド9の内容で指定された値予測器12の予測値が出力され、値予測フィールド8の内容が‘1’のとき、予測値が値予測マスクゲート14からレジスタ16及び値予測結果判定器17に出力される。レジスタ16の予測値は演算器15で後続の命令の処理に用いられる。判定器17において、演算器15による命令の処理結果と予測値が比較され、不一致のとき、キャッシュメモリ中の命令の値予測フィールド8と値予測方式フィールド9のデータを更新する。

【選択図】 図 6

出 願 人 履 歴 情 報

識別番号 [000005108]

1. 変更年月日 1990年 8月31日

[変更理由] 新規登録

住 所 東京都千代田区神田駿河台4丁目6番地

氏 名 株式会社日立製作所